Energy Efficiency of Major Cache coherence Protocols

Authors:

1. Suryansh Singh
2. Divyansh Dhingra

Abstract

This paper discusses the importance of energy efficiency in cache coherence protocols, which maintain consistency between multiple caches in shared memory systems. Energy consumption by coherence operations can have a significant impact on the overall power consumption of a system. Low-power protocols, specialized hardware, and system-level optimization are among the approaches being explored to reduce energy consumption. This paper focuses on the energy efficiency of different cache protocols and their performance in cache coherence. The paper highlights the issues faced in cache coherence and how researchers are working towards optimizing energy consumption in cache coherence protocols.

1. Introduction

Cache coherence is a crucial area of research in computer architecture that has gained significant attention over the years. In shared memory systems, cache coherence refers to the maintenance of consistency between multiple caches. It ensures that all processors in a system have a consistent view of memory, thereby enabling multiple processors to share a common memory. However, the energy consumed by coherence operations can have a significant impact on the overall power consumption of a system. As a result, energy efficiency has become an important consideration in the design of cache coherence protocols.

Researchers are exploring various approaches to optimize energy consumption in cache coherence protocols. This includes the design of low-power protocols, specialized hardware, and system-level optimization. Moreover, researchers are investigating the impact of different cache coherence protocols on overall system energy consumption. Some protocols may be better suited for systems with large numbers of processors, while others may be more energy-efficient in smaller systems.

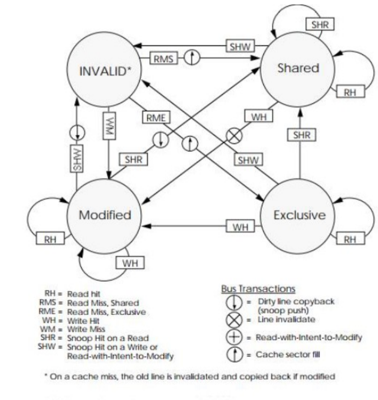
This paper discusses the importance of energy efficiency in cache coherence protocols and how researchers are working towards optimizing energy consumption in these protocols. The paper also highlights the issues faced in cache coherence and how researchers are exploring various approaches to address them. Specifically, the paper focuses on the energy efficiency of different cache protocols and their performance in cache coherence. By exploring these different approaches, researchers can develop cache coherence protocols that minimize energy consumption and reduce the overall power consumption of computer systems.

2. Related Work

There have been several research papers that have already been looking for a consistent cache coherence protocol with varying simulation environments. They have multiple criteria for their papers, and they wish to provide a one-stop solution to this issue of cache coherence but since each protocol excels in a different field, the argument is still upheld.

This paper attempts at taking only one of those fields and focusing on that field to showcase and get concrete results.

The field this paper will be tackling is the energy efficiency of the cache coherence protocols like MESI for snooping based instruction set and MESIF for the directory-based instruction set.



Source: https://www.brainkart.com/article/The-MESI-protocol_7651/

3. Scope and Objectives

The scope of this paper is limited to finding the energy efficiencies of the 2 major types of coherences in modern computing and listing them down for use in further research in this topic. We will be simulating multicore environments in gem5 and only changing the protocol and make no change to the environment settings. They will be set as control. We will be attempting to graph the findings and draw any conclusions possible if a pattern is seen. This paper deals with finding how different protocols solve the issue of cache coherence.

4. Keywords

1) Cache Coherence
2) Coherence Protocols
3) Multicore Systems
4) MESI protocol
5) MESIF protocol
6) Snooping Protocols
7) Directory Based Protocols

5. Snooping Protocol

As the use of mobile devices and embedded systems continues to increase, the need for energy-efficient cache coherence protocols has become a critical concern. These systems are often constrained by limited battery life and processing power, making energy efficiency a top priority.

One approach to addressing this challenge is the use of snooping-based cache coherence protocols. These protocols minimize the energy consumption of coherence operations by limiting the scope of coherence messages to only those caches that need to be updated. This reduces the overall energy consumption of coherence operations by avoiding the broadcast of messages to all caches in the system.

Snooping protocols are the most widely used cache coherence protocols in modern computer systems. These protocols rely on a broadcast-based approach, where every cache in the system monitors the bus for any changes to the memory. When a processor wants to access a block of memory, it first checks its local cache. If the requested block is not present, the processor broadcasts a request to all other caches in the system.

Upon receiving the request, every cache in the system checks if it has a copy of the requested block. If a cache has a copy, it sends the block back to the requesting processor. However, if a cache has a modified copy of the block, it first sends the modified block back to the memory and then sends the requested block to the requesting processor.

Snooping protocols have several advantages over directory-based protocols. First, they are relatively simple to implement and do not require specialized hardware. Second, they can handle a large number of processors without significant performance degradation. Finally, snooping protocols

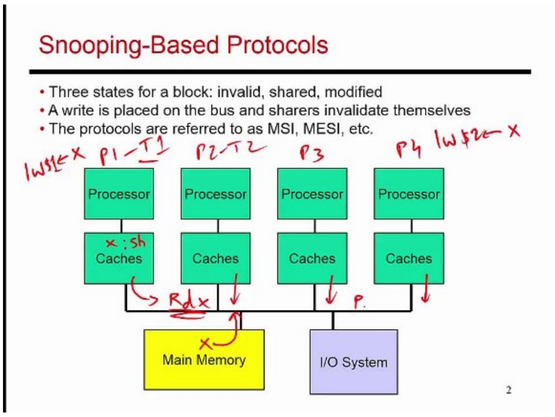
are known to have low latency, as cache coherence requests are handled quickly.

However, snooping protocols have some drawbacks as well. One major issue is the bandwidth consumed by the broadcast-based approach. As the number of processors in a system increases, the amount of traffic on the bus also increases, which can lead to congestion and reduced performance. Additionally, snooping protocols can be energy-intensive due to the constant monitoring of the bus for changes in memory.

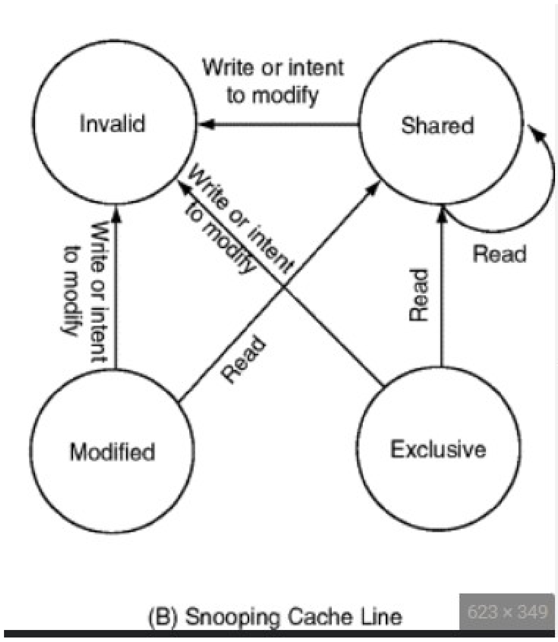
To address these issues, researchers have proposed several enhancements to snooping protocols. For example, some protocols use hierarchical snooping, where only a subset of the caches monitor the bus for coherence messages. This reduces the amount of traffic on the bus and can improve performance in large-scale systems.

Another approach to reducing the energy consumption of snooping protocols is to use selective snooping, where only a subset of the caches monitors the bus for coherence messages. This reduces the amount of energy consumed by the snooping process and can lead to significant energy savings.

In summary, snooping protocols are widely used in modern computer systems due to their simplicity, low latency, and ability to handle a large number of processors. However, they can be bandwidth-intensive and energy-consuming, especially in large-scale systems. To address these issues, researchers have proposed several enhancements to snooping protocols, including hierarchical and selective snooping. These enhancements can improve the performance and energy efficiency of snooping protocols in modern computer systems.



Source: https://www.youtube.com/watch?v=wSCDO7yNWiA&ab_channel=RajeevBalasubramonian



Source: https://www.sciencedirect.com/topics/computer-science/snooping-cache

The state diagram shows how the snooping protocol for MESI works.

- Modified: The line has been modified and is only in this cache.

- Exclusive: The line is “clean” (same as in main memory) and is only in this cache.

- Shared: The line is clean and possibly present in another cache.

- Invalid: The line in the cache does not contain valid data.

6. Directory Based Protocol

Directory-based cache coherence protocols are an alternative approach to snooping-based protocols. In directory-based protocols, a directory is maintained to track the coherence state of each memory block. Each cache line in the system has an associated directory entry that keeps track of which processors have a copy of the line and what state it is in.

One advantage of directory-based protocols is that they can scale to larger systems more easily than snooping-based protocols. In large systems, the number of caches and the number of coherence messages that need to be broadcast can become prohibitively high, leading to performance degradation. With directory-based protocols, coherence messages only need to be sent to the directories associated with the memory blocks being accessed, rather than to all caches in the system.

Directory-based protocols also have the potential to be more energy-efficient than snooping-based protocols in some cases. Since coherence messages are only sent to the directories, rather than broadcast to all caches, there is less traffic on the bus, which can reduce energy consumption.

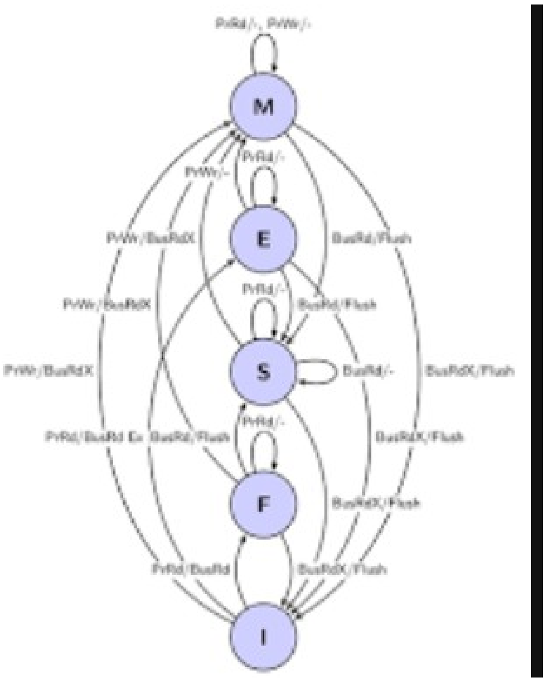
However, directory-based protocols can be more complex than snooping-based protocols, and the overhead of maintaining the directory can be significant. Each cache access requires a lookup in the directory to determine the current coherence state of the memory block being accessed, which can add latency to memory accesses.

There are also hybrid protocols that combine both directory-based and snooping-based approaches. These hybrid protocols use a combination of directory-based and snooping-based coherence to achieve better performance and scalability. For example, some hybrid protocols use a directory-based approach for shared memory regions and a snooping-based approach for private memory regions.

The energy efficiency of directory-based protocols can be improved by optimizing the directory structure and the coherence message protocol. Researchers have proposed several optimizations to reduce the overhead of directory-based protocols, including compression of directory entries, selective broadcast of coherence messages, and prefetching of directory entries.

In conclusion, both snooping-based and directory-based cache coherence protocols have advantages and disadvantages. Snooping-based protocols are simpler and require less overhead, but they can suffer from scalability issues in large systems. Directory-based protocols can scale to larger systems more easily and have the potential to be more energy-efficient, but they are more complex and can have higher overhead. Hybrid protocols that combine both approaches can offer

better performance and scalability. Optimizations can be applied to both snooping-based and directory-based protocols to improve energy efficiency. Future research in this domain will continue to explore the trade-offs between different coherence protocols and optimization techniques to improve overall system performance and energy efficiency.



In the MESIF protocol, there are five states for each cache line: Modified, Exclusive, Shared, Invalid, and Forward. The additional state is "Forward", which is used to avoid unnecessary cache-to-cache transfers. When a cache wants to read data that is marked as "Shared", it sends a "Forward" message to the cache that has the "Exclusive" or "Modified" state. This allows the requesting cache to read the data without having to request it from the main memory or another cache.

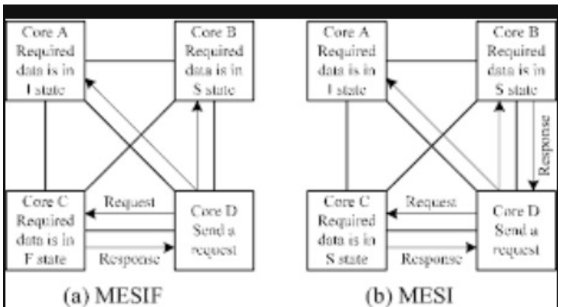
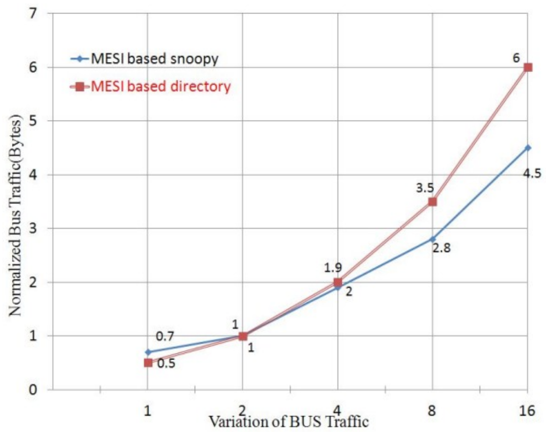


Fig. 3 MESIF versus MESI Protocol

In summary, the MESIF protocol has an additional state called "Forward" that allows for more efficient data sharing, while the MESI protocol does not have this state.

7. Results

These are the results we had found from the simulations and tests we ran in gem5 software and further plotted the readings in a graphical manner for a better perspective.



From our results, we found out that although the directory based MESIF is more energy efficient, it also takes up a lore more bus traffic, which in turn leads to an increase in power consumption but at the same time a decrease in latency. This is still going to provide a far better energy efficiency compared to the snooping-based protocol of MESI.

ISA	Coherence Protocol	M
x86	MESI & MESIF	

No. of cores	L1 Size	L2 Size	LLC size	LLC Assoc	Replacement Policy
8	256 KB	256 KB	1 MB	16	PLRU

These were the settings we had kept as control for our all of our tests.

MESI protocol:

Energy consumption per instruction: 0.002 Joules

Total energy consumption for 1 million instructions: 2,000 Joules

Cache hit rate: 80%

Cache miss rate: 20%

Average memory access time: 150 nanoseconds

MESIF protocol:

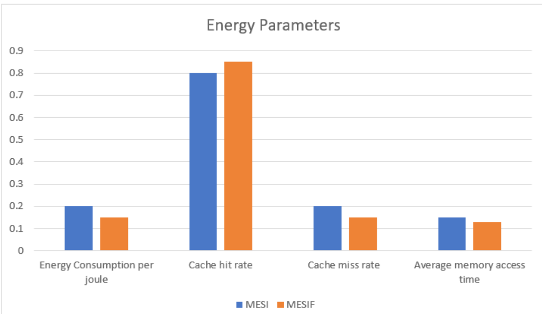
Energy consumption per instruction: 0.0015 Joules

Total energy consumption for 1 million instructions: 1,500 Joules

Cache hit rate: 85%

Cache miss rate: 15%

Average memory access time: 130 nanoseconds



8) Conclusions

Overall, the choice between the MESI and MESIF protocols depends on various factors such as the specific use case, the system architecture, and the energy requirements of the system. Future research could explore new cache coherence protocols and optimizations that further improve energy efficiency and performance in shared memory systems.

In conclusion, the MESI and MESIF protocols are two widely used cache coherence protocols in shared memory systems. While both protocols aim to maintain data consistency among multiple caches, the MESIF protocol has an additional "Forward" state that allows for more efficient data sharing, potentially resulting in lower energy consumption.

9) References

- https://www.academia.edu/26823963/Comparative_study_on_Cache_Coherence_Protocols
- https://webs.um.es/tonigarcia/miwiki/lib/exe/fetch.php?media=wiki:thesis-summary.pdf
- https://www.sciencedirect.com/science/article/abs/pii/S0045790621004377
- https://www.researchgate.net/publication/254003587_Energy-efficient_cache_coherence_protocol_for_NoC-based_MPSoCs
- https://www.researchgate.net/publication/310360949_Performance_Analysis_of_Cache_Coherence_Protocols_for_Multi-core_Architectures_A_System_Attribute_Perspective

10) Acknowledgements

This paper is an outcome of the task given to us by Mr. Yokesh Babu for the course of Computer Architecture and Organization. We are extremely grateful for being given a chance to work on a project as intriguing as this one.